ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Title goes here

Master Thesis

Robin Bosshard, 16-915-399

October 16, 2023

Supervisors: Prof. Dr. Fernando Perez-Cruz, Dr. Eliza Harris, Lili Gasser (SDSC)
Dr. Kasia Arturi (Eawag)

Department of Computer Science, ETH Zürich

Contents

Chapter 1

Background

This chapter provides information essential for comprehending the subsequent sections of this thesis. It begins by introducing the challenges and the evolving trends in toxicity testing. Following that, we explore the ToxCast's *invitro* database together with its processing pipeline, *tcpl*. These concepts are crucial for acquainting ourselves with the process of generating the respective toxcity data from the *high-throughput screening (HTS)* used for the underlying predictive machine learning models.

1.1 Toxicity Testing: From In Vitro Assays and Molecular Fingerprints to Predictive Models and Beyond

With the ever-growing amount of chemical compounds entering our environment, conducting experiments on all these compounds using traditional methods have constraints related to expense and time, and ethical considerations regarding animal trials.

In 2007, the *U.S. National Academy of Sciences* introduced a visionary perspective and published a landmark report, titled as *Toxicity Testing in the 21st Century: Vision and Strategy*. This report promoted a transition from conventional, resource-consuming animal-based *in vivo* tests to efficient high-throughput *in vitro* pathway assays on cells.

This transition paved the way for the realm of high-throughput screening (HTS), where a multitude of *in vitro*bioassays can be executed, complementing and improving chemical screening. This transformation is made possible by advancements in robotics, data processing, and automated analysis. As a result, this synergy has led to the generation of extensive toxicity datasets like ToxCast and Tox21.

HTS datasets, including *ToxCast* and other sources, have opened the door to promising applications of machine learning in predictive computational

toxicology. PThese predictive models can be developed to screen environmental chemicals with limited toxicity data availability, allowing for the prioritization of further testing efforts. Such models often forecast toxicity using *Quantitative Structure-Activity Relationships (QSARs)*, which are based on chemical structures encoded as descriptors, such as molecular fingerprints. 1D-Molecular fingerprints encode compound molecules as fixed-length binary vectors, denoting the presence (1) or absence (0) of specific substructures or functional groups.

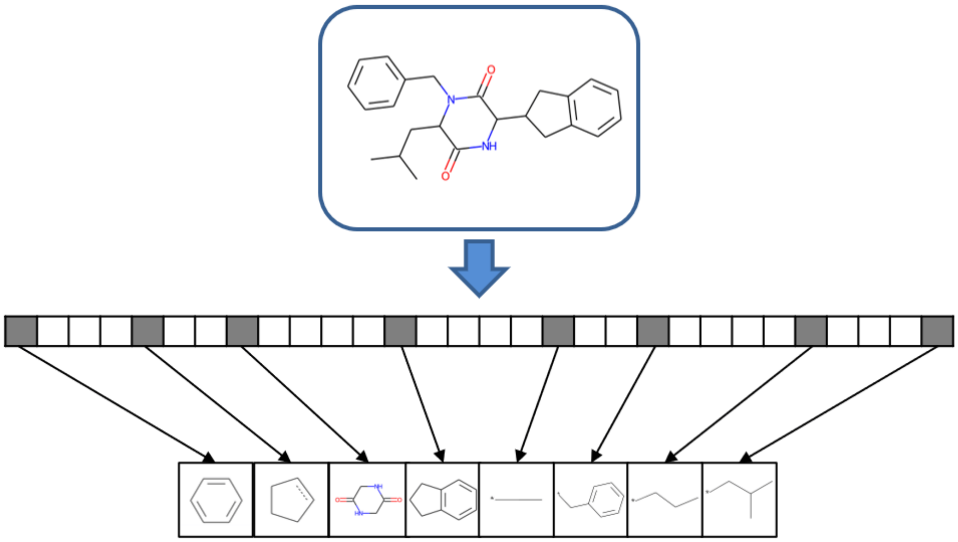


Figure 1.1: Figure 1 adapted from Janela et al (2022) [1]. Schematic molecular fingerprint. Each bit position accounts for the presence or absence of a specific structural fragment. Bit positions are set on (set to 1, gray) if the substructure is present in a molecule, or set off (set to 0, white) if it is absent.

The utilization of molecular fingerprints for in vitro toxicity prediction is based on the assumption that molecular toxic effects result from relatively straightforward interactions between distinct chemical components and receptors during a *molecular initiating event (MEI)*. On a larger biological scale, the *MEI* can set a sequential chain of causally linked *key events (KE)* in motion. This occurs at different levels of biological organisation from within cells to potentially culminating in an *adverse outcome pathway (AOP)* at the organ or organism level, as depicted in Figure 1.2. The mechanistic information captured in *AOPs* reveal how chemicals or other stressors cause harm, offering insights into disrupted biological processes, potential intervention points but also guide regulatory decisions on next generation risk assessment and toxicity testing. The *AOP* framework is an analytical construct that allows an activity mapping from the presence or absence of certain molecular sub-

structures encoded in chemical descriptors to the target mechanistic toxicity. Finally, when monitoring disruptions in toxicity pathways, *physiologically based pharmacokinetic (PBPK)* models can be leveraged to extrapolate *in vitro* findings to human blood and tissue concentrations [2].

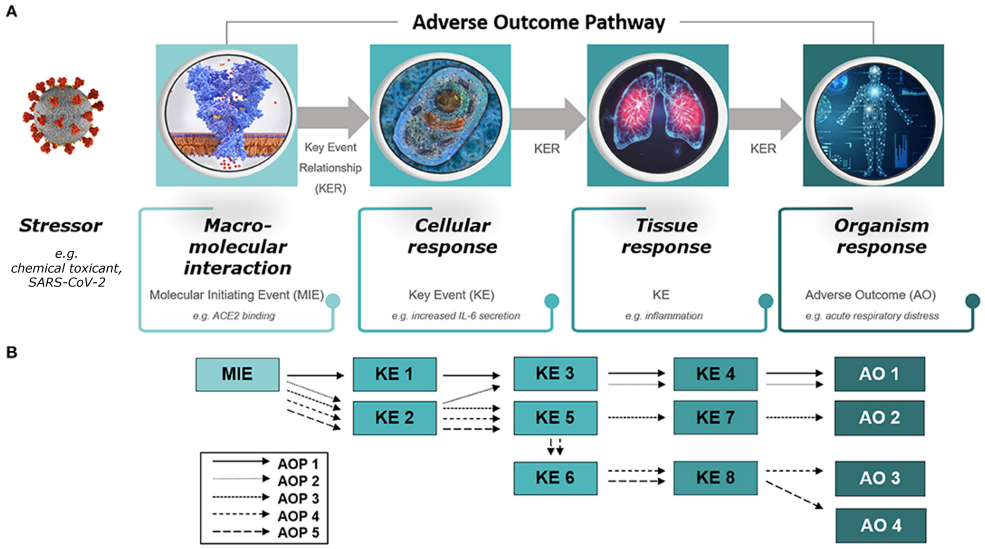


Figure 1.2: Figure 1 adapted from [3]: Diagram of (A) an adverse outcome pathway (AOP) and (B) an AOP network. (A) An AOP starts with a molecular initiating event (MIE), followed by a series of key events (KEs) on different levels of biological organization (cellular, tissue, organ) and ends with an adverse outcome (AO) in an organism. The stressor is not part of the AOP itself.

1.2 Chemical Target Toxicity vs. Cytotoxicity

Intuitively, we expect increasing chemical concentrations to result in increasing bioactivity. However, this is not always the case. At higher doses the chemical can become cytotoxic leading to dying cells. In consequently, a reduction in bioactivity can occur, e.g., the activation of the reporter gene decreases.

Chemical toxicity can manifest in diverse ways, falling into two major categories [4]:

- **Specific toxicity** is the result of a chemical's interaction and disruption of a specific biomolecular target or pathway, such as a receptor agonist/antagonist effect or enzyme activation/inhibition.
- **Cytotoxicity and cell stress** is the generalized disruption of the cellular machinery. Cell-disruptive processes encompass various mechanisms, such as protein, DNA, or lipid reactivity, or processes like apoptosis,

oxidative stress responses or mitochondrial disruption. Cell viability can be evaluated either individually or concurrently. One approach is to assess it by calculating the proportion of live cells in a population, employing a fluorescent dye that specifically enters living cells. This dye remains incapable of permeating the membranes of deceased cells, resulting in fluorescence intensity directly correlating with cell viability.

It is common for compounds to exhibit target bioactivity within a limited concentration range, which coincides with a non-specific activation response in the presence of cell stress and cytotoxicity. Figure 1.3 illustrates the interference between specific toxicity and cytotoxicity.

A related phenomenon is referred to as the *cytotoxicity burst* [4],is observed, where, for example, reporter genes may be induced non-specifically near the point of cell death [5]. The *ToxCast* pipeline is designed to minimize false negatives by adopting an inclusive risk assessment approach. However, due to interference processes, the reliability of reported activities becomes uncertain, and false positives can occur without a thorough cytotoxicity evaluation. While some of the assay activity within this concentration range may indeed result from chemical interactions with the intended assay target, another portion does not and needs to be taken into account.

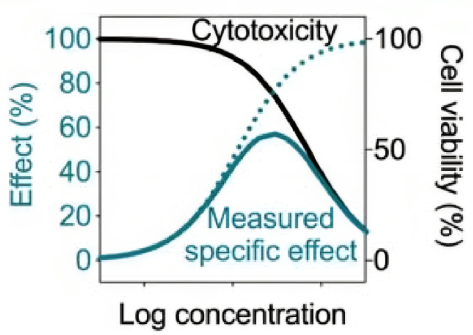


Figure 1.3: Figure 7.8 from Escher at al. [5]: Bioanalytical Tools in Water Quality Assessment: Second Edition. Example of a bioassay response with cytotoxicity interference. The dotted line shows the theoretical effect but due to cytotoxicity (black line is cell viability), the measured effect has an inverted U-shape. The measured effect can addionally be confounded and intensified by the cytotoxicity burst, where even an exponential shape is likely for the gaining part. In this case, the effect should be only evaluated up to some concentration.

1.3 InvitroDB v4.1

The most recent release of the *ToxCast's (Toxicity Forecaster)* database[1], referred to as *invitroDBv4.1*, serves as a source of an extensive collection of *HTS*

[1]released on September 21, 2023

targeted bioactivity data. This database encompasses information on a total of 10 196 compounds, selectively screened across 1485 assay endpoints. The assays utilize a range of technologies to assess the impact of chemical exposure on a wide array of biological targets, including individual proteins and cellular processes such as mitochondrial health, developmental processes and nuclear receptor signaling.

This resource originated from the collaboration of two prominent institutions: the *United States Environmental Protection Agency*[2] *(EPA)* through its *ToxCast* program and the *National Institutes of Health*[3] *(NIH)* via the *Tox21*[4] initiative. Utilizing data gathered from various research laboratories, this relational database is publicly available and can be downloaded[5] by visiting the official *ToxCast* website.

1.4 tcpl v3.0

The *tcpl* package, written in `R`, offers a comprehensive suite of tools for managing HTS data, provides reproducible concentration-response modeling and populates the `MYSQL` database, *invitrodb*. The multiple-concentration screening paradigm intends to pinpoint the activity of compounds, while also estimating their efficacy and potency. The concentration-response modeling procedure also addresses outlier robustness and signal loss due to cytotoxicity. In Chapter 2, the `Python` re-implementation *pytcpl* of the fundamental components of the ToxCast pipeline *tcpl* is introduced but at this point the essential elements are layed out that underpin the entire pipeline.

Each compound-assay dataset involves the collection of the respective *concentration-response series (CRS)* denoted as S_{ij}, representing compound c_j in assay endpoint a_i. A CRS is represented as a set of concentration-response pairs:

$$S = \{(conc_1, resp_1), (conc_2, resp_2), \ldots, (conc_{n_{\text{datapoints}_{i,j}}}, resp_{n_{\text{datapoints}_{i,j}}})\}$$

where $n_{\text{datapoints}_{i,j}}$ varies based on the number of concentrations tested for compound c_j in the assay endpoint a_i.

The concentration-response pairs can be retrieved by combining tables *mc0*, *mc1*, and *mc3* from *invitroDBv4.1*, representing the raw data. Esential sample information, including well type and indices from the assay well-plate is also collected. The concentrations are transformed to the logarithmic scale having the unit μM (micromolar), while the responses are control well-normalized to either fold-induction or percent-of-control activity. A control well is a

[2] https://www.epa.gov
[3] https://ntp.niehs.nih.gov/whatwestudy/tox21
[4] https://tox21.gov/
[5] https://www.epa.gov/chemical-research/exploring-toxcast-data

set of sample wells that serve as a baseline for comparison to the impact of the treated samples. Control wells typically contain untreated samples or samples with a known, non-toxic response. The control wells are used to normalize the treated samples to account for any background noise or variability in the assay. There are two methods for analyzing raw assay results that will impact the analysis of the background distribution [6]:

a. **Fold Induction:** Fold induction is a measure used to quantify how much a certain parameter (e.g., gene expression or protein activity) has changed in response to a treatment compared to its baseline level. For example, if a gene is expressed five times higher in a treated sample compared to the control, the fold induction would be 5.

a. **Percent of Control:** Percent of control is another way to express the relative change in a parameter due to treatment where the obervations range from 0 to a maximum value for both chemical-treated and control samples.

Figure 1.4: A CRS for the compound *Estropipate* (DTXSID3023005) in the assay endpoint *TOX21_ERa_LUC_VM7_Agonist* (aeid=788). The series has a total of $k = 45$ concentration-response pairs and is composed of $n_{conc} = 15$ concentration groups, each with $n_{rep} = 3$ replicates.

In practice, concentrations are often subjected to multiple testing iterations, resulting in the formation of distinct concentration groups with replicates. The following quantities are introduced corresponding to a single CRS given a compound c_i and assay endpoint a_i:

- $n_{\text{datapoints}_{i,j}}$: the total number of concentration-response pairs ($|S|$)
- $n_{\text{groups}_{i,j}}$: the number of distinct concentrations tested
- $n_{\text{replicates}_{i,j}}$: the number of replicates for each concentration group
- $min_{\text{conc}_{i,j}}$: the lowest concentration tested

- $max_{\text{conc}_{i,j}}$: the highest concentration tested

Figure 1.4 showcases a single CRS for some compound tested within an assay endpoint.

To gain a rough visual representation of how these quantities vary across the complete set of analyzed concentration-response series in this thesis, please consult Figure 1.5. This figure illustrates the above metrics aggregated by their means, grouped by assay endpoints and compounds.

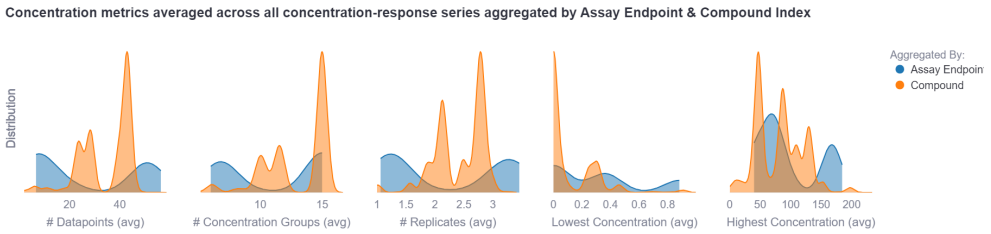


Figure 1.5: Concentration metrics averaged across all concentration-response series aggregated by assay endpoint (blue) and compound (orange). E.g., the first chart shows the distribution on the average number of datapoints across all assay enpoint $a_i \in A$ with $\frac{1}{|C|}\sum_j n_{\text{datapoints}_{i,j}}$ and across all compounds $c_j \in C$ with $\frac{1}{|A|}\sum_i n_{\text{datapoints}_{i,j}}$. Similarly, the process is repeated for the other metrics: $n_{\text{groups}_{i,j}}$, $n_{\text{replicates}_{i,j}}$, $min_{\text{conc}_{i,j}}$, and $max_{\text{conc}_{i,j}}$.

1.4.1 Tcplfit2

To improve upon tcpl, `R` package *tcplfit2* was developed, a standalone package focused on curve-fitting and hit-calling. The package also offers a more flexible and robust fitting procedure, allowing for the use of different optimization algorithms and the incorporation of user-defined constraints. Tcplfit2 differs from other R-language open-source concentration-response packages like *drc* and *mixtox*, as it is specifically tailored for HTS concentration-response data, offering an extensive set of curve models, summarized in Table 1.1.

All the curve fit models assume that the normalized observations from the CRS conform to a Student's t-distribution with 4 degrees of freedom [7] The Student's t-distribution has heavier tails compared to the normal distribution, making it more robust to outlier and eliminates the necessity of removing potential outliers prior to the fitting process. The model fitting algorithm in `tcplFit2` employs nonlinear *maximum likelihood estimation (MLE)* to determine the model parameters for all available models.

Consider $t(z, \nu)$ as the Student's t-distribution with ν degrees of freedom, where y_i represents the observed response for the i-th observation, and μ_i is the estimated response for the same observation. The calculation of z_i is as follows:

Table 1.1: tcplfit2 Model Details

Model	Label	Equations[1]
Constant	cnst	$f(x) = 0$
Linear	poly1	$f(x) = ax$
Quadratic	poly2	$f(x) = a\left(\frac{x}{b} + \left(\frac{x}{b}\right)^2\right)$
Power	pow	$f(x) = ax^p$
Hill	hill	$f(x) = \frac{tp}{1+\left(\frac{ga}{x}\right)^p}$
Gain-Loss	gnls	$f(x) = \frac{tp}{(1+\left(\frac{ga}{x}\right)^p)(1+\left(\frac{x}{la}\right)^q)}$
Exponential 2	exp2	$f(x) = a\left(\exp\left(\frac{x}{b}\right) - 1\right)$
Exponential 3	exp3	$f(x) = a\left(\exp\left(\left(\frac{x}{b}\right)^p\right) - 1\right)$
Exponential 4	exp4	$f(x) = tp\left(1 - 2^{-\frac{x}{ga}}\right)$
Exponential 5	exp5	$f(x) = tp\left(1 - 2^{-\left(\frac{x}{ga}\right)^p}\right)$

[1] Model parameters: a: x-scale, b: y-scale p: (gain) power, q: (loss) power, tp: top, ga: gain AC50, la: loss AC50

$$z_i = y_i - \mu_i \exp(\sigma)$$

where σ is the scale term.

Then the log-likelihood is

$$\sum_{i=1}^{n} [\ln(t(z_i, 4)) - \sigma]$$

where n is the number of observations.

The model with the lowest *Akaike Information Criterion (AIC)* value is selected as the *winning* model. The winning model is then used to estimate the efficacy and potency of the compound. More precise, the potency estimates, also called *point-of-departure (POD)* estimates, are derived from the fitted curve characteristics, identifying *activity concentrations (AC)* at which the curve crosses certain response levels. For example:

- the AC at which the compound first reaches 50% of its estimated maximum response is denoted by *ac50*

- the AC at which the compound first reaches the estimated efficacy cutoff is denoted by *acc*
- the AC at which the compound first reaches the estimated assay background noise *3bmad*[6] is denoted by *acb*

Figure X illustrates the stated *POD* estimates. Notably, no *POD* estimates are computed when the compound is considered inactive[7], as these estimates are not applicable in such cases.

The *continuous hitcall*[8] is calculated based on the product of the probabilities of the following values [6]:

i. that at least one median response is greater than the efficacy cutoff computed by using the error parameter from the model fit and Student *t*-distribution to calculate the odds of at least one response exceeding the efficacy cutoff

ii. that the top of the winning fitted curve is above the cutoff:
the likelihood ratio of the one-sided probability of the efficacy cutoff being exceeded

iii. that the winning AIC value is less than that of the constant model:

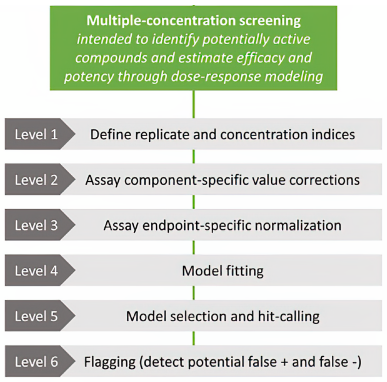
$$\frac{e^{-\frac{1}{2}AIC_{\text{winning}}}}{e^{-\frac{1}{2}AIC_{\text{winning}}} + e^{-\frac{1}{2}AIC_{\text{cnst}}}}$$

Finally, after processing, each CRS is categorized into an appropriate fit category based on the level of certainty in the estimated bioactivity. Additionally, cautionary flags are assigned to account for problematic data series or uncertain fits and hits.

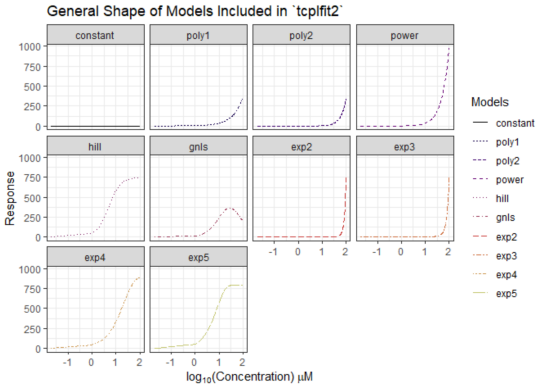
[6]The baseline region is defined as $0 + 3bmad$, where *bmad* represents the median absolute deviation calculated from the response values of the two lowest tested concentrations, where the assumption of the highest level of inactivity is legitimate.

[7]if the *winning* fit model was the constant model

[8]In legacy tcpl only binary hitcall was calculated



(a) Overview of tcpl's data analysis pipeline for multiple concentrations screening data.



(b) Employed curve-fit models in tcpl v3.0 for fitting concentration-response data series through the application of maximum likelihood estimation.

Figure 1.6: tcpl

Chapter 2

Material and Methods

2.1 Data Overview

Presence Matrix

Consider a collection of m assay endpoints, denoted by $A = \{a_1, a_2, \ldots, a_m\}$ and a set of n compounds represented as $C = \{c_1, c_2, \ldots, c_n\}$. To facilitate data comprehension, we introduce a *presence matrix* $P \in \{0,1\}^{m \times n}$. Rows, indexed by i, represent assay endpoints a_i, while columns, indexed by j, denote presence (1) or absence (0) of compound c_j in those endpoints. Matrix P is sparse due to the selective testing of compounds across different assay endpoints. A compound is considered present in an assay endpoint if it has undergone testing and a corresponding concentration-response series is available. See Figure 2.1 for a visual of the *presence matrix P* covering all assay endpoints and compounds in *invitroDBv3.5*.

Subsetting data

We exclusively consider assay endpoints that have been tested with a minimum of 2000 compounds. This criterion ensures the availability of sufficient data for the training of a machine learning model. Refer to Figure 2.2 for a visual representation of the *presence matrix P*, which now encompasses only the resulting subset of all assay endpoints within *invitroDBv3.5*. From now on, we will call this specific subset the data that we will be focusing on for this thesis.

2.2 Pytcpl

We introduce pytcpl, a streamlined Python package inspired by the R package tcpl, designed for processing high-throughput screening data. The package primarily focuses on providing essential features such as concentration-

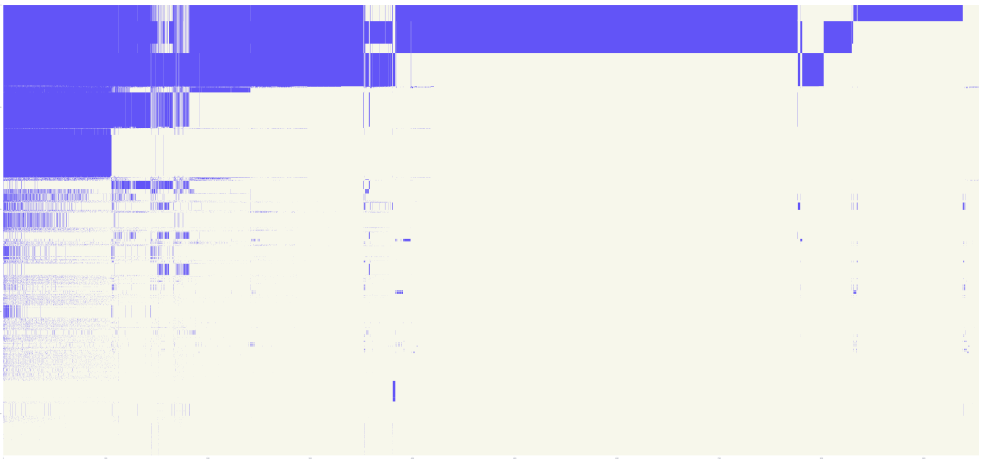


Figure 2.1: The *presence matrix* P covering all assay endpoints and compounds available in *invitroDBv3.5* with $m = 2205$ assay endpoints and $n = 9541$ compounds. The presence matrix is organized by sorting it based on the number of compounds present in each assay endpoint and the compounds are arranged in descending order of their presence frequency. The total count, where $P_{ij} = 1$, indicates the availability of 3 342 377 concentration-response series for downstream analysis.

response curve fitting and allows for continuous hit-calling for compound bioactivity across diverse assay endpoints, akin to tcplfit2. Invitrodb version 3.5 release can optionally serve as backend database if desired. The package optimizes data storage and provides compressed raw data and metadata from *invitroDB* in Parquet files. This efficient strategy reduces storage needs, resulting in just 4 GB within the repository—compared to the original 80 GB database. This obviates the need for a cumbersome, large-scale database installation, rendering downstream analysis more accessible and efficient. Our package is crafted to accomodate cusomizable processing steps and facilitate interactive data visualization with curve surfer. Moreover, it empowers Python-oriented researchers to seamlessly engage in data analysis and exploration.

We analyse in total $\sum_{i,j} P_{ij} = 1\,372\,225$ concentration-response series, comprising a sum of $\sum_{i,j} |S_{ij}| = 48\,861\,036$ concentration-response pairs across all compounds and assay endpoints.

2.2.1 Pipeline

1. Data collection
2. Cutoff determination and filtering (Meet conditions for curve fitting)
3. Curve fitting

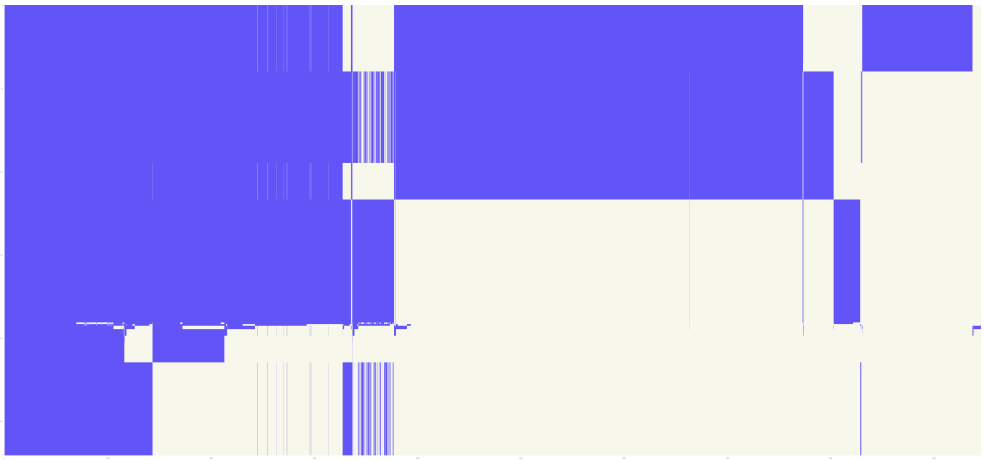


Figure 2.2: The *presence matrix* P covering only the subset of all of assay endpoints available in *invitroDBv3.5*, considered for this thesis, encompassing $m = 271$ assay endpoints and $n = 9456$ compounds. The total count, where $P_{ij} = 1$, indicates the availability of 1 372 225 concentration-response series for downstream analysis.

4. Hit calling

Data Collection

First, all datapoints are collected from the database and assigned to the concentration response-series belonging to the respective compound in the corresponding assay endpoint.

Curve Fitting

Introduce all candidate fit models, discuss the pros and cons of each model. Discuss the fitting procedure, how the models are fitted, Maximum Likelihood Estimation.

Model	Label	Equations[1]	Role in pytcpl
Exponential 3	exp3	$f(x) = a\left(\exp\left(\left(\frac{x}{b}\right)^p\right) - 1\right)$	Omit
Gain-Loss 2	gnls2	$f(x) = \frac{tp}{1+\left(\frac{ga}{x}\right)^p} \exp\left(-qx\right)$	New

[1] Model parameters: a: x-scale, b: y-scale p: (gain) power, q: (loss) power, tp: top, ga: gain AC50

Table 2.1: tcplfit2 Model Details

Hit Calling

Akaike criterion, probability of being active, etc..

$$AIC = -2\log(L(\hat{\theta}, y)) + 2K \tag{2.1}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.2}$$

2.2.2 Curve Surfer

Data visualization, overview of what is possible with the tool. Filter by assay endpoint, compound, etc.

2.3 Machine Learning Pipeline

2.3.1 Preprocessing

Subselecting the columns from the output tables generated by pytcpl: DTXSID identifier and continuous hitcall value. The feature inputs to the machine learning model is a molecular structure represented as fingerprint generated from a SMILES string uniquely determined by the compounds DTXSID identifier. The SMILES string is a linear representation of a compound's molecular structure. The SMILES string is converted to a molecular graph, which is then converted to a feature vector. The feature vector is then used to train a machine learning model. The machine learning model is then used to predict the hitcall value for a given compound. The machine learning pipeline is illustrated in Figure?.

2.3.2 Binary Classification

The goal is to predict whether a compound is active or inactive for a given assay endpoint. We can formulate this as a binary classification problem, where the input is the compound's molecular structure fingerprint and the output is the hitcall value binarized by some decision threshold. The hitcall value is rendered to a binary variable, where 1 indicates that the compound is active and 0 indicates that the compound is inactive.

2.3.3 Regression

2.3.4 Massbank Validation

Bibliography

[1] T. Janela, K. Takeuchi, and J. Bajorath, "Introducing a chemically intuitive core-substituent fingerprint designed to explore structural requirements for effective similarity searching and machine learning," *Molecules*, vol. 27, no. 7, 2022, ISSN: 1420-3049. DOI: 10.3390/molecules27072331. [Online]. Available: https://www.mdpi.com/1420-3049/27/7/2331.

[2] S. M. Bell *et al.*, "In vitro to in vivo extrapolation for high throughput prioritization and decision making," *Toxicology in Vitro*, vol. 47, pp. 213–227, 2018, ISSN: 0887-2333. DOI: https://doi.org/10.1016/j.tiv.2017.11.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0887233317303661.

[3] P. Nymark *et al.*, "Systematic organization of covid-19 data supported by the adverse outcome pathway framework," *Frontiers in Public Health*, vol. 9, May 2021. DOI: 10.3389/fpubh.2021.638605.

[4] R. Judson *et al.*, "Editor's Highlight: Analysis of the Effects of Cell Stress and Cytotoxicity on In Vitro Assay Activity Across a Diverse Chemical and Assay Space," *Toxicological Sciences*, vol. 152, no. 2, pp. 323–339, May 2016, ISSN: 1096-6080. DOI: 10.1093/toxsci/kfw092. eprint: https://academic.oup.com/toxsci/article-pdf/152/2/323/26290632/kfw092.pdf. [Online]. Available: https://doi.org/10.1093/toxsci/kfw092.

[5] B. Escher, P. Neale, and F. Leusch, *Bioanalytical Tools in Water Quality Assessment*. IWA Publishing, Jun. 2021, ISBN: 9781789061987. DOI: 10.2166/9781789061987. eprint: https://iwaponline.com/book-pdf/899726/wio9781789061987.pdf. [Online]. Available: https://doi.org/10.2166/9781789061987.

[6] T. Sheffield, J. Brown, S. Davidson, K. P. Friedman, and R. Judson, "tcplfit2: an R-language general purpose concentration–response modeling package," *Bioinformatics*, vol. 38, no. 4, pp. 1157–1158, Nov. 2021, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab779. eprint: https:

//academic.oup.com/bioinformatics/article-pdf/38/4/1157/50422999/btab779.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/btab779.

[7] C. for Computational Toxicology and U. E. Exposure, *Tcpl v3.0 data processing*, R package vignette for the tcpl package v3.0, CRAN, 2023. [Online]. Available: https://cran.r-project.org/web/packages/tcpl/vignettes/Data_processing.html.